UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/706,625	11/06/2000	David Francis Bacon	YOR920000359US2	8377

7590 11/14/2006

THOMAS A. BECK, ESQ
26 ROCKLEDGE LANE
MILFORD, CT 06776

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 11/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/706,625	BACON ET AL.	
	Examiner	Art Unit	
	Li B. Zhen	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 August 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,5 and 7-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,5 and 7-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 5 and 7 – 16 are presented for examination.

Response to Arguments

2. Applicant's arguments filed 08/19/2004 have been fully considered but they are not persuasive. In response to the Non-Final Office Action dated 07/02/2003, applicant argues:

(1) Examiner's assertion with respect to Ungar's system heap "not being a means of garbage collection" is clearly refuted at Column 5, line 30 etc. which states: "System heap 31 2...contains proxy objects to help in the garbage collection process..." Thus the Examiner's position in this respect is erroneous and, accordingly, the rejection is without foundation. [p. 5, lines 3 – 9];

(2) The present invention and the Ungar reference are antithetical as far as their teachings are concerned as Applicants do not collect garbage in either the system heap or the transient heap and Ungar does collect garbage in the system heap and the transient heap. [p. 5, lines 9 – 11];

(3) O'Toole does not describe a middleware heap in his disclosure because the "persistant heap" of O'Toole is analogous to the "system heap" defined by Applicants. [p. 5, lines 12 – 16];

(4) There is no proper basis to combine the references Ungar, O'Toole and Printezis references as Ungar implements garbage collection in his system heap and O'Toole has no middleware heap. [p. 6, lines 11 – 16]; and

(5) Examiner in the Official Action has selected concepts improperly from the references out of context as the basis for the rejections. [p. 7, lines 9 – 15].

In response to argument (1), examiner respectfully disagrees and submits that column 5, line 30 of Ungar discloses "System heap 312 includes catalog 314, which contains proxy objects to help in the garbage collection process for application heap 310" [emphasis added]. The garbage collection process is for the application heap and not the system heap; therefore, Ungar does not disclose that the system heap is garbage collected. Although the application heap is garbage collected, the application heap is not garbage collected until execution of the application is completed [col. 5, lines 13 – 25 of Ungar]. Claim 1 recites that the transient heap is not garbage collected within the lifetime of the application. Ungar teaches that the data objects are reclaimed when they are no longer referenced and when the data objects are not referenced, the application is no longer running. Therefore, Ungar's system heap meets the recited system heap and Ungar's application heap meets the recited transient heap.

As to argument (2), examiner disagrees and notes that applicant's claims require garbage collection in the transient heap [claim 15, lines 13-14 and claim 16, lines 15-16; "performing garbage collection on said transient heap"]. Although claim 1 recites that the transient heap is not garbage collected within the lifetime of the application, the transient heap is cleared in between successive applications. Therefore, the transient heap is garbage collected in between successive applications. Ungar's application heap is also garbage collected in between successive applications [system must

determine which data objects within application heap 310 are being referenced, and which data objects can be reclaimed; col. 5, lines 13 – 25]. Ungar's garbage collection process is for the application heap and not the system heap [column 5, lines 30-33]; therefore, the system heap of Ungar is not garbage collected. Ungar's system heap meets the recited system heap and Ungar's application heap meets the recited transient heap.

As to argument (3), examiner disagrees and submits that the persistent heap of O'Toole meets the recited middleware heap. The persistent heap of O'Toole meets the recited middleware heap because the persistent heap is garbage collected [garbage collection of the persistent heap; p. 7 – 8, Persistent GC] as recited in applicant's claims [claim 1, line 8; claim 15, line 11; claim 16, line 13]. Examiner disagrees with applicant's submission that the "persistent heap" of O'Toole is analogous to the "system heap" defined by applicants because O'Toole's persistent heap is garbage collected and applicant's system heap is not garbage collected. Since O'Toole's persistent heap is garbage collected, the persistent heap meets the recited middleware heap which is also garbage collected.

In response to argument (4), examiner disagrees and notes that the Office Action provides proper basis to combine the references of Ungar, O'Toole and Printezis [see rejection to claim 8 below]. As to applicant's arguments that Ungar implements garbage collection in his system heap and O'Toole has no middleware heap, see the response to arguments (1) – (3) above.

As to argument (5), examiner disagrees and submits that the cited concepts from the references meet applicant's recited claims.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1, 5, 7, 15, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over in view of U.S. Patent No. 6,275,985 to Ungar in view of "Concurrent Compacting Garbage Collection of a Persistent Heap" [hereinafter O'Toole], both references cited in the previous office action.

5. As to claim 1, Ungar teaches the invention substantially as claimed including an object-based virtual machine environment [virtual machine 116, Fig. 1; column 3, lines 60 – 67], storage for storing objects for running the applications [storage system 120, Fig. 1; column 3, lines 60 – 67], said storage being logically divided into heaps [col. 5, lines 1 – 10]:

a system heap wherein system classes for said single virtual machine are loaded into the system heap, thereby providing subsequent applications with the ability to used these classes without having to reload them [system heap 312 includes memory for use by the system that executes the application...execution of the system causes a number

of objects to be allocated and manipulated in system heap 312; column 5, lines 25 – 30] and which system heap is not garbage collected [system heap 312, Fig. 3; column 5, lines 1 – 45]; and

a transient heap [application heap 310, Fig. 3; column 5, lines 1 – 45] which is cleared inbetween successive applications, and which has no garbage collected within the lifetime of an application [During garbage collection, the system must determine which data objects within application heap 310 are being referenced, and which data objects can be reclaimed; col. 5, lines 13 – 25] said transient heap being used for storing applications objects that are used for only the duration of an application [During execution of an application, virtual machine 116 maintains a stack 302 for the application as well as an application heap 310; col. 5, lines 1 – 10]. Ungar does not appear to teach a middleware heap.

However, O'Toole teaches an object-based system with middleware [automatic storage management in transaction systems, object-oriented databases and persistent programming environment; p. 1, Abstract], a transient heap [Transitory Heaps for Temporary Data; p. 2 – 3, Design; p. 5, Section 4.1 The Transitory Heap] that is cleared inbetween successive applications [transitory heap contains only temporary objects that will be discarded when a failure occurs; p. 5, Section 4.1 of O'Toole], and which has no garbage collected within the lifetime of an application [p. 7, Section 4.6], and a middleware heap [Persistent Heap; p. 6 – 7, Section 4.4 The Persistent Heap] which is garbage collected [garbage collection of the persistent heap; p. 7 – 8, Persistent GC].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to apply the teaching of a middleware heap as taught by O'Toole to the invention of Ungar because this provides object persistence, and safe and efficient transactional storage [p. 1, see Abstract].

6. As to claim 15, Ungar as modified teaches a method of operating a computer system providing an object-based virtual machine environment [virtual machine 116, Fig. 1; column 3, lines 60 – 67 of Ungar], in which middleware runs successive applications on a single virtual machine [automatic storage management in transaction systems, object-oriented databases and persistent programming environment; p. 1, Abstract of O'Toole], said system including storage for storing objects for running the applications [storage system 120, Fig. 1; column 3, lines 60 – 67 of Ungar], said method comprising the steps of:

logically dividing the storage into three heaps [col. 5, lines 1 – 10 of Ungar];

a system heap wherein system classes for said single virtual machine are loaded into the system heap, thereby providing subsequent applications with the ability to used these classes without having to reload them [system heap 312 includes memory for use by the system that executes the application...execution of the system causes a number of objects to be allocated and manipulated in system heap 312; column 5, lines 25 – 30 of Ungar];

a middleware heap [Persistent Heap; p. 6 – 7, Section 4.4 The Persistent Heap of O'Toole];

a transient heap [application heap 310, Fig. 3; column 5, lines 1 – 45 of Ungar];

performing garbage collection on said middleware heap between successive applications [garbage collection of the persistent heap; p. 7 – 8, Persistent GC of O'Toole]; and wherein reusable objects other than class objects that must persist between successive applications are stored in said middleware heap [promote all newly persistent data into volatile from-space; p. 7, Section 4.6 of O'Toole] and performing garbage collection on said transient heap [During garbage collection, the system must determine which data objects within application heap 310 are being referenced, and which data objects can be reclaimed; col. 5, lines 13 – 25 of Ungar], but not on said system heap [system heap 312, Fig. 3; column 5, lines 1 – 45 of Ungar]; and

clearing the transient heap inbetween successive applications, and which has no garbage collected within the lifetime of an application [During garbage collection, the system must determine which data objects within application heap 310 are being referenced, and which data objects can be reclaimed; col. 5, lines 13 – 25 of Ungar] said transient heap being used for storing applications objects that are used for only the duration of an application [During execution of an application, virtual machine 116 maintains a stack 302 for the application as well as an application heap 310; col. 5, lines 1 – 10 of Ungar] and which has no garbage collected within the lifetime of an application [p. 7, Section 4.6 of O'Toole].

7. As to claim 16, this is a product claim that corresponds to method claim 15; note the rejection to claim 15 above, which also meets this product claim.

8. As to claim 5, Ungar as modified teaches only portion of the storage corresponding to the middleware heap is garbage collected between successive applications [garbage collection of the persistent heap is initiated; p. 7, Persistent GC of O'Toole].

9. As to claim 7, Ungar as modified teaches any objects in the transient heap that are eligible for use by the next application, and which are referenced by live objects in the system heap or middleware heap, are promoted to the middleware heap [update all of the transitory heap data to point to the newly promoted objects...this is done either by scanning the transitory heap form pointers to promoted objects; p. 7, Section 4.6 of O'Toole].

10. **Claims 8 – 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ungar and O'Toole in view of U.S. Patent No. 6,249,793 to Printezis [cited in the previous office action].**

11. As to claim 8, Ungar as modified does not teach a card table.

However, Printezis teaches [column 6, lines 30 – 50] a card table [a data structure 70 referred to as a "card table"], in which each card corresponds to a portion of the middleware heap [card table 70 includes entries associated with memory regions within the heap 40...each entry in the card table 70 includes information regarding the

memory region it is associated with], and the card is marked if the middleware heap potentially references an object in the transient heap [a card table entry may include a dirty bit, which when set indicates that one or more pointers within the associated memory region have been modified since the dirty bit was last cleared].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to apply the teaching of a card table as taught by Printezis to the invention of Ungar as modified because a card table entry may indicates that one or more pointers within the associated memory region have been modified or store information that includes time stamps reflecting a time at which a pointer in the associated region was last modified [column 6, lines 40 – 50 of Printezis].

12. As to claim 9, Ungar as modified teaches each card corresponds to a memory region having a size greater than the minimum size for an object [card table 70 includes entries associated with memory regions within the heap 40...each entry in the card table 70 includes information regarding the memory region it is associated with; column 6, lines 30 – 50 of Printezis].

13. As to 10, Ungar as modified teaches a card is marked whenever an object in the corresponding memory region is updated [a card table entry may include a dirty bit, which when set indicates that one or more pointers within the associated memory region have been modified since the dirty bit was last cleared; column 6, lines 30 – 50 of Printezis].

14. Claims 11 – 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ungar and O'Toole in view of U.S. Patent No. 5,950,008 to van Hoff [cited in the previous office action].

15. As to claim 11, Ungar and O'Toole does not teach a middleware class loader and an application class loader.

However, van Hoff teaches a generating application specific class loaders [column 5, lines 23 – 38].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to apply the teaching of generating application specific class loaders as taught by van Hoff to the invention of Ungar as modified because each application specific class loader contains information such as location information, fetch method for fetching object classes, linking method and load method that are specific to each class [column 5, lines 30 – 38 of van Hoff]. Obviously, a middleware specific class loader can be created to load middleware class objects.

16. As to claim 12, Ungar as modified teaches objects from classes loaded by the one or more system class loaders are created in the middleware heap or the transient heap depending on the current context [application specific class loader 300 itself contains both a class reference 302 and a data reference 304 to a data array 305; column 5, lines 23 – 40 of van Hoff].

17. As to claim 13, Ungar as modified teaches the current context is middleware if the method being run derives from a class loaded by the middleware class loader, and application if the method being run derives from a class loaded by the application class loader [application specific class loaders do not include an application specific class loader generator method 312, because it is assumed that all symbol references to be resolved by the application specific class loader will be successfully resolved by locating an associated object class on either the server identified by the location information 303; column 6, lines 1 – 10 of van Hoff].

18. As to claim 14, Ungar as modified teaches if the method being run derives from a class loaded by the one or more system class loaders, then the current context retains the value it had immediately before the method was run [a new application specific class loader is generated for that called method by the generator method...the new application specific class loader is then used to load the object class on that other server as well as to locate and load the object classes for any method calls made by that method; column 6, lines 10 – 41 of van Hoff].

Conclusion

19. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

CONTACT INFORMATION

20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2194

LBZ

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 21[illegible]